UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/735,715	12/12/2000	Jacob Dreyband	033144-004	5590

23562 7590 10/17/2005

BAKER & MCKENZIE
PATENT DEPARTMENT
2001 ROSS AVENUE
SUITE 2300
DALLAS, TX 75201

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 10/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/735,715	DREYBAND ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 July 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21 and 43-71* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21 and 43-71* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 July 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 7/29/2005.

As indicated in Applicant's response, claims 1, 43 have been amended. Claims 1-21, 43-71 are pending in the office action.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1-21, 43-71 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Specifically, both claims 1 and 43 recite " machine language transferable among, and directly processible by, without any intermediate data format conversions, any computer processing unit processing data for use by any application operating in any computer environment, platform, architecture or language' (*). There is no support in the disclosure in regard to elements (disclosed as a whole or individually) constituting the packing as binary representation of tagged data in a *machine language* directly processible without any intermediate data format conversion by any processing unit for use *by any application*. Scanning the entire disclosure, it is found that there is no machine level representation being disclosed such that it would be construed as being directly processible without any intermediate data

format conversion (emphasis added) for use by any application (emphasis added). In other words, there is absence of a minimal description relating how such machine level representation is packed then directly processible (e.g. there is complete absence of the term *directly processible*) for use by ANY application -- there's only reciting of the term *application* as in a API, hence no mention of a application layer usage without any trace of manipulation by an interface code-- in ANY computer environment, platform, architecture or language (e.g. except for a may-be container). Paragraph 32 and 59 mention about a tagged data formatted in wire format by means of some programming language *emp.pack()* function to yield a structure like a array that *may be* (see also para 11) a container that is platform independent, hardware architecture independent and language independent. Disclosing a binary form having packed data as a wire format such that it *may be* a container that is platform independent, hardware architecture independent and language independent does not amount to the limitation recited as in (*), especially when there is no explicit mention about absence of any intermediate data format conversion as claimed so that the packed data can be used by ANY computer application (when the term *application* as commonly understood is not even disclosed – Note: an underlying API is not a application per se in useful arts related to user application). Data being transferable in binary representation was a known concept and data being processed from different layers of a network protocol so that it become a format suitable for application use was also a known process. Disclosing that binary representation at level of wire format is communicated to and directly used at the application level without a conversion requires clear teaching from the disclosure to distinguish it from known concepts. A container (e.g. *may be a container*) that is platform independent, hardware architecture independent and language independent does not

convey a possession of the subject matter thus claimed. One skill in the art would find it very hard to construe the claimed limitation from the disclosure, i.e. from the representation in wire format being implemented from a programming code packing constructs and then possibly being a container (that is platform independent, hardware architecture independent and language independent) to a point at which this representation can be directly processed by any processing unit for use in ANY application without any intermediate data format conversion, when there is no explicit description about how this intermediate conversion has been bypassed (or how the absence of conversion has been implemented) such that ANY application can use via direct processing, while the platform, architecture, language independency is based on the teaching of a 'may-be' container. Hence, the above limitation (as in *) as a whole is considered not possessed by the inventor at the time of the invention; and the limitation would be interpreted as broadly as reasonably allowed.

Dependent claims 2-42, and 44-71 are also rejected for not remedying to the deficiencies of the base claims.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-21, and 43-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah, USPN: 6,477,580 (hereinafter Bowman), in view of Francis et al., USPN:

6,665,861 (hereinafter Francis), and further in view of White et al., 6,438,559 (hereinafter White).

As per claim 1, Bowman discloses a method for presenting data within a computing environment including an application program interface (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object for storing data (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187; Fig. 98 – Notes: browser interpreting pages is equivalent to tagged data being stored in message or packages streamed between browser applications or interconnected framework machines);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 98; Fig. 184-185);

packing the tagged data by converting the tagged data as a binary representation of tagged data object (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191- Note: a stream being passed over the internet reads on binary representation of being packed tagged data being streamed and eventually processed by recipient machine application engines).

But Bowman does not explicitly specify that the package or message of tagged data are storing universal tagged data object being platform independent, hardware independent, and language independent. But in view of Bowman's disclosing of COM format for effecting RPC, messaging utilities and directory services having platform independent standard for transmitting data (e.g. Fig. 20-22; col. 73, lines 10-41; col. 63, line 62 to col. 63, line 21 – Note: COM format

data are platform and language neutral by nature of common platform object broker services), in combination of language neutral for Java byte codes (*virtual machines 2706* – Fig. 27), and binary representation across hardware independent internet protocol, the above limitation is at least strongly suggested. The packaging of data using Java platform neutral format was a well-known concept in the art of software transmission at the time the invention was made. Francis, in a method to transmit package of Java binary representation and metadata similar to Bowman streaming of data (Bowman: Fig. 108-109) across computers, also discloses packaging binary representation of Java beans with supporting utilities/metadata under markup or tagged form like XML (Fig. 6-8). In case the platform, hardware and language neutral package stream by Bowman are not universal tagged data for browser use, it would have been obvious for one of ordinary skill in the art at the time the invention was made to encapsulate such data in XML form as taught Francis because this will alleviate resources of the receiving computer in making use of readily formatted data without additional compilation.

Further, Bowman does not explicitly disclose universal tagged data being encapsulated for universal manipulation and aggregation by computer processing units of tagged data; however, the very fact of having stream of binary packets across platform discloses encapsulated data for universal access, universal in the context of many machines that can establish reception of such stream; hence Bowman disclose universal tagged data. As for the access to manipulation and aggregation of tagged data, Bowman discloses browser manipulation of tagged data and markup language data manipulation and aggregation using browser application in conjunction with stream, message passing and ORB remote calls using platform independent-based services as mentioned above (e.g. Fig. 13-18; Fig. 98); hence has implicitly disclosed access of tagged (

or markup) data for manipulation or aggregation (Note: data provided through COM or ORB services and a compilation of HTML formatted data in pages composed of subdivided markup sections implicitly disclose access for manipulation or aggregation of data).

Nor does Bowman explicitly disclose tagged data object capable of being transferred among and directly processible by, without any intermediate data format conversions, any computer processing unit processing data for use by any application operating in any computer environment, platform, architecture or language. Bowman discloses web format aggregating inherent layers of markup tags, markup data (e.g. Fig. 13-18; Fig. 24, 98), such data being sent turn into a binary stream format for enabling the transmission over different computers through platform independent messaging services or protocols as mentioned above for browser processing without recompilation. This in combination with the rationale as set forth above using Francis' teachings discloses tagged data object (i.e. a binary stream representing a tagged data) being transferred and processed without any intermediate format conversions because browser can make use of markup language as received in browser applications.

But Bowman does not explicitly specify that the encapsulated tagged data provides data that includes data element and a corresponding binary tag id. Bowman or Francis, however, teaches tagged data for interpretation by browsers, hence implicitly discloses a variable name bracketed within the begin tag and end tag; and teaches attributes descriptors in the meta-data section comprising a header section of the object-based stream, which suggests encapsulating identification information of the data element of the stream, according to a well-known concept of including some identifier in a binary form to associate the data bundle or packet sent over a network communication link with its content at the time the invention was made. White, in a

method to serialize objects for distribution over a communication network environment using descriptors in serialization of class objects analogous to the object streaming and meta-data by Bowman or Francis, discloses the tagging of object content being serialized with an identifier or value for packing and deserializing (e.g. *ACI* - col. 4, lines 16-58; col. 10, line 36 to col. 11, line 9). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the tagging technique associating an identifier with the tagged content as taught by White and apply it to the stream metadata by Bowman, in case Bowman's metadata or tagged stream does not include such tag identification already, because this tag ID would facilitate the differentiation between data being packed and enable data handling/re-processing as well as unpacking or modification of elements packed in the message or bundle.

As per claim 2, Bowman discloses the packing of tagged data being a simple object and a complex object, and list object (e.g. col. 124, line 14 to col. 127, line 39 – Note: the use of Java or C++ based components implicitly discloses basic class, compound classes, or structure/enumeration of basic classes and compound classes objects).

As per claim 3, Bowman discloses packing a simple object by retrieving data attributes for length of an object source identifier, object size, type, value; allocating of packed memory location for object identifier length (e.g. col. 235, line 47 to col. 237, line 32); copying the object size, type, and value into the packed memory location (Note: this is inherent to the above cited portions); retrieving and copying head value and exit value into the packed memory location (e.g. *START INDEX, WS-INDEX, STREAM-END* - col. 237, line 35 to col. 238, line 66).

As per claims 4 and 5, Bowman does not explicitly specify the steps of retrieving, writing, and allocating/writing for the complex object as has been disclosed for the simple object

from claim 3, but in view of the packaging of data in the retrieval of business-related complex object (e.g. col. 204, line 40 to col. 207, line 59), the limitations as recited are herein implicitly in view of the inherent presence of simple object within complex object or list objects.

As per claim 6 and 7, Bowman does not specify packing list object with retrieving of object source identifier, allocating memory in a packed memory location to accommodate the list object source identifier length; retrieving and copying list head value and list exit value into the packed memory location; but in view of the rationale used in addressing claims 4 and 5, these limitations are also implicitly disclosed because of the inherent presence of simple objects and complex objects in structure or enumeration, i.e. list, object so well-known in object-oriented language.

Further, Bowman does not explicitly disclose retrieving list array object and copying it to the packed memory location. But, in view of the inherent array structure in structure or enumeration of simple and complex objects in C++ or Java, this limitation is also implicitly disclosed as per the same rationale used for claims 4 and 5.

As per claim 8, Bowman does not explicitly disclose that the tagged data object is an universal data container that is platform, language, and architecture independent for access to manipulation and aggregation of structured or unstructured data; but in view of rationale used in claim 1 to address tagged data being hardware, platform and language independent and provided for access to manipulation, aggregation tagged data, the limitation is rejected herein with the same rationale as set forth therein (Note: structure data is formatted data stream according to HTML, XML or internet or COM protocol reads on container; as opposed to unstructured data are generic data used or referenced indirectly by such structured data)

As per claim 9, see Bowman (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191).

As per claim 10, refer to claim 2.

As per claim 11, Bowman discloses data wrapping (e.g. *Wrapper component* - Fig. 81).

As per claim 12, Bowman (col. 131-132; col. 174, line 33 to col. 175, line 24; Fig. 50-51), discloses modeling using COM and Case Tools but Bowman does not specify including of named tree with a field name connected with a value. But in view of the teaching of language-independent modeling along with metadata or language neutral format as addressed in claim 1, (Francis' teaching modeling and tagged web format data is suggesting of tree structure implementation of data to be transmitted as metadata or specification data), this limitation would have obvious for the same rationale as used in claim 1 and also the association of tree with field name as metadata would enhance the utilization and re-processing of data tagged and stored in the package.

As per claim 13, Bowman discloses Java and C++ constructs which inherently include list or enumeration of objects of simple and complex type (see claim 2).

As per claim 14, this claim includes the encapsulation of data type, tag id, and writing thereof to the tagged data object and these limitations have been addressed in claim 3 and 4.

As per claim 15, Bowman discloses the use of Java objects, hence has implicitly disclosed one of the following data type: integer, float, byte, char string, a java object, a null data, a primitive type, a compound type, and a list type.

As per claim 16, Bowman (in combination with White/Francis) discloses tag identifier with of type integer (see White from claim 1).

As per claim 17, Bowman with White's teachings discloses serializing of tagged data and compacting it in a stream for transmission, hence has implicitly disclosed a tagging process following a linear sequence, i.e. sequential tagging with determining a sequence.

As per claim 18, Bowman with White's teachings discloses including a data, a position and a tag element (refer to claim 3; Fig. 109 – Note: Index position use in writing data by Bowman discloses including an position and packet layout inherently encompasses boundaries position of data compacted in packet).

As per claim 19, Bowman does not explicitly specify converting of first type of tagged data to second type of data for a change in properties; but the concept for converting the order of data type (e.g. network-bound integer converted into local host-based integer and vice-versa, as per Java/C++ *ntohs* or *htons* functions) for allowing data type to be communicated through the internet medium was a well-known concept at the time the invention was made. Hence, Bowman's disclosed communication of Java or C++ objects implicitly discloses such conversion to provide for a communication properties adjustment or change as claimed.

As per claims 20 and 21, by virtue of the rejections of claim 2 and claim 19 above, the limitations of these claims are implicitly disclosed.

As per claim 43, Bowman discloses a method for presenting data within a computing environment including an application program interface prescribed for data conversion and wire formatting specification (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187); wherein the tagged data object comprises a universal data container

that is platform and hardware independent (e.g. col. 99, lines 7-40 – Note: Java platform independency is implicitly disclosed); said tagged data object providing broad access to manipulation and aggregation of structured data and unstructured data (e.g. *view configurator, maximum maintainability and extensibility* - col. 248, line 28 to col. 259, line 45; *LUW* - Fig. 108-129, 163-191 – Note: context retrieving and selecting appropriate objects from requests is equivalent to broad access for data manipulation and aggregation);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 184-185);

providing the tagged data by converting the tagged data as a binary representation of tagged data object (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191);

transmitting the tagged data transmission (e.g. Fig. 105-107);

unpacking the tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29).

But Bowman does not explicitly disclose tagged data object to provide universal access to manipulation and aggregation by computer processing units of a structured data and unstructured data; but this limitation has been addressed in claim 1 and 8 above.

Nor does Bowman explicitly disclose that the packed tagged data object is capable of being transferred among and directly processible by, without any intermediate data format conversions, any computer processing unit processing data for use by any application operating in any computer environment, platform, architecture or language. As mentioned above, Bowman

discloses web format aggregating inherent layers of markup tags, markup data (e.g. Fig. 13-18; Fig. 98), such data being sent turn into a binary stream format for enabling the transmission over different computers through platform independent messaging services or protocols as mentioned above for browser processing without recompilation. This in combination with the rationale as set forth above using Francis' teachings discloses tagged data object (i.e. a binary stream representing a tagged data) being transferred and processed without any intermediate format conversions because browser can make use of markup language as received in browser applications.

Nor does Bowman explicitly specify that the encapsulated tagged data object includes a data element and a corresponding tag id; but this also has been addressed in claim 1 above using Francis/White.

As per claims 44-49, these claims correspond to claims 2-7 respectively; hence are rejected likewise, respectively.

As per claim 50, this corresponds to claim 2, and is rejected using the rationale of claim 2. Further, Bowman discloses a method for presenting data within a computing environment including an application program interface comprising the steps of unpacking tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29-- Note: in view of the teachings on packing data into package or stream to be sent in packet over the internet by Bowman as mentioned in claim 1, the steps of unpacking, creating a storage for the unpacked data received over the internet, and the extracting of object being tagged are implicitly disclosed).

As per claim 51, Bowman does not specify the steps of retrieving the simple head value and simple exit value; allocating memory in an unpacked memory and copying of simple object size, type and value into said unpacked memory. Official notice is taken that subjecting packets received from the internet into a host or routing, or a gateway machines to unpacking and buffer storage was a well-known concept in the art at the time the invention was made. In view of the teachings for unpackaging of data by Bowman above and the well-known unpacking of data, it would have been obvious for one skill in the art at the time the invention was made to provide the unpacking of the tagged data as taught by Bowman/Francis/White using the well-known technique of unpacking/storage above because this would enable correct extraction of data based on boundaries locations and allocation of correct memory resources.

As per claims 52 and 53, the limitations as to unpack a complex object would also have been obvious by virtue of the inherency of simple object in a complex objects as mentioned in claims 4-5 and the rejection used in claim 51 above.

As per claims 54 and 55, the rationale used for claims 6-7 and 52-53 are herein applied.

As per claims 56-59, refer to rejections of claims 10-13 respectively.

As per claim 60, this claim corresponds to claim 14, hence is rejected using the same rationale as set forth therein.

As per claims 61-67, refer to corresponding rejections of claims 15-21 respectively.

As per claim 68, Bowman does not specify extracting data with determining the type to provide the tag id; and writing the data element into the tagged data object. But in view of White's or Francis's teaching to provide a tagging associated with an identifier in order to facilitate the reprocessing of data manipulated at the receiving end and the rationale for

encapsulating in claim 14, this step would have been obvious because the implied and inherent association between packing and unpacking.

As per claims 69 and 70, see rejection of claims 15 and 16 respectively.

As per claim 71, in view of the unpacking as taught by Bowman and the rationale in claim 18 above, this limitation would also have been obvious by virtue of the adding of element in the tagged data as mentioned in the above rejection.

Response to Arguments

6. Applicant's arguments with respect to mainly claims 1, 43 have been considered but are not persuasive. Following are the Examiner's corresponding counter arguments and observations thereto.

(A) Applicants have submitted that Bowman merely discloses 'browser that employ HTML and XML formats to process data they receive ... both of which ... textual formats ... not machine level ... suitable for immediate processing ... for application level programs' (Appl. Rmrks, pg. 16, bottom, pg. 16, pg. 17, 1st para) and that Bowman does not disclose a universal tagged object ... universal manipulation and aggregation ... processing units ... for use by any application in ... computer environment, platform, architecture or language'. As mentioned in the USC 112, 1st paragraph rejection, there is sufficient teaching from the disclosure for the claim to be perceived as having patentable weight just as recited. The limitation as to universally manipulated and aggregated has been addressed in the rejection 1) as a format which any application can incorporate and make use - XML markup language neutral specification data for processing; and 2) as binary stream for transmission for being processed by different type of environment --regardless of architecture and programming language--via low layers of parsing

packets and their header. It is noted that the entire limitation as specified in the USC 112, 1st paragraph rejection does not amount to viable and/or substantial patentable weight because of lack of disclosure; and thus has been treated by means of broadest interpretation without undue analysis based on one skill in the art when compelled to interpret with insufficient insight. Therefore, the arguments on Bowman's converting into textual format are non persuasive when the claimed invention is silent as to what this improved technique of bypassing (not using at all) any intermediate conversion amounts to particularly when the specifications mention of an interface code (API) to process the tagged data.

(B) Applicants have submitted that like Bowman, Francis discloses a textual format. This argument is based on the validity allotted to a properly claimed limitation; which is not the case as per the 35 USC 112, 1st paragraph (Appl. Rmrks, pg. 16, bottom, pg. 17, middle) from above. The argument about Francis' conversion into textual format falls under the ambit of Bowman's textual format being addressed above. Besides, Francis is used with Bowman in a rationale based on a combination for obviousness. Applicants apparently have failed to provide a convincing argument showing why the combination as put forth in the rejection would be inappropriate or generate adverse effects. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

For the above reasons, the claims stand rejected as set forth above.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
October 7, 2005

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100